

# A Low-Computation-Cycle Design of Input-Decimation Technique for RIDFT Algorithm

Chih-Feng Wu, <sup>1</sup>Chun-Hung Chen and <sup>2</sup>Muh-Tian Shiue

Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan.

<sup>1</sup>Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan.

<sup>2</sup>Department of Electrical Engineering, National Central University, Taoyuan 32001, Taiwan.

tfeng.wu@gmail.com; henrikchentw@gmail.com; mtshiue@ee.ncu.edu.tw;

**Abstract**—In this paper, a low-computation-cycle and energy-efficient design of input-decimation technique for the recursive inverse discrete Fourier transform (RIDFT) algorithm is proposed for the high-speed broadband communication systems. It is crucial that the input-decimation technique is presented to decrease the number of input sequences for the recursive filter so that the computation cycle of RIDFT can be shortened to meet the computing time requirement (3.6  $\mu$ s). Therefore, the input-decimation RIDFT algorithm is able to carry out at least 55.5% reduction of the total computation cycles compared with the considered algorithms. Holding the advantages of input-decimation technique, the computational complexities of the real-multiplication and -addition are reduced to 41.3% and 22.2%, respectively. Finally, the physical implementation results show that the core area is  $0.37 \times 0.37$  mm<sup>2</sup> with 0.18  $\mu$ m CMOS process. The power consumption is 5.16 mW with the supply voltage of 1.8 V and the operating clock of 40 MHz. The proposed design can achieve 258 million of computational efficiency per unit area (CEUA) and really outperform the previous works.

**Index Terms**—recursive inverse discrete Fourier transform (RIDFT), orthogonal frequency-division multiplexing (OFDM)

## I. INTRODUCTION

The recursive discrete Fourier transform (RDFT)/inverse DFT (RIDFT) has been widely used in many fields of digital signal processing including dual-tone multi-frequency (DTMF) detection [1]–[3] and digital radio mondiale (DRM) receiver [4]–[9]. These works [2][3][5]–[9] had presented the RDFT/RIDFT designs with low computation cycle, low computational complexity and high area efficiency in the last decade. The module-sharing and the register-splitting schemes for RDFT/RIDFT design [2] were presented to reduce the number of multipliers and to cut the critical path, respectively. Van *et al.* [3] revised Goertzel formula [10] to offer an RDFT/RIDFT architecture constructed by a pre-processing, a recursive discrete cosine transform and a recursive sine transform for yielding the computation cycle of  $N^2/2$ . Depending on the module-sharing and the register-splitting, Lai *et al.* [5] rewrote Goertzel formula to present an RDFT/RIDFT architecture with symmetric property for achieving the computation cycle of  $(N/2 - 1) \cdot (N + 1)$ . Based on the kernel of [5] with the factor decomposition, a series of the RDFT/RIDFT evolutions were presented such as the dual-RDFT [6] (2-RDFT

kernels), the multi-RDFT [7] (3-RDFT kernels), the folded-RDFT [8] (an iterative operation of single RDFT kernel) and the hybrid architecture [9] (a configurable hardware module used for RDFT and radix-2<sup>2</sup> FFT).

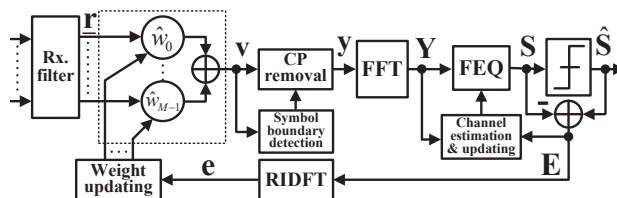


Fig. 1. Beamforming technique using RIDFT for broadband communication systems [11].

Considering the beamforming tracking for IEEE 802.11n OFDM receiver [11] as shown in Fig. 1, an RIDFT situated at the backward path fulfills the decision error transformation from the frequency domain to the time domain for the weight updating of beamforming. The major motivation for applying RIDFT to do the beamforming tracking in OFDM receiver is described as (1) No need an extra inverse FFT (IFFT) placed at the backward path, (2) Real-time processing without input storage during the packet reception, and (3) Execution time of RIDFT  $\leq 3.6$   $\mu$ s since an OFDM symbol duration with the normal/short guard interval is 4.0/3.6  $\mu$ s. Referring to the specifications of DTMF [1] and DRM/DRM<sup>+</sup> [4], the requirement of the execution time is exactly from 2.5 to 40 ms. Therefore, these designs [6][7][8][9] (80~384  $\mu$ s) can not be applied to the beamforming tracking for IEEE 802.11n OFDM receiver.

In this paper, an input-decimation technique is proposed to derive RIDFT algorithm and then to yield the low-computation-cycle and the energy-efficient design of RIDFT for beamforming tracking of OFDM receiver. This paper is organized as follows. The proposed RIDFT algorithm using the input-decimation technique and the symmetric property is derived in Section II. The hardware design is described in Section III. Both computational- and hardware-complexity of the presented input-decimation RIDFT algorithm compared with the considered algorithms [3][5]–[9] are discussed in Section IV. Finally, the conclusions are given in Section V.

This work was supported in part by Ministry of Science and Technology, Taiwan, under Grant MOST 104-2218-E-346-001, 106-2622-8-008-006-TA, 106-2221-E-008-099-MY3 and 107-2622-8-008-005-TA.

## II. PROPOSED RECURSIVE IDFT ALGORITHM

It is crucial that the input-decimation technique is proposed to minimize the number of input sequences of recursive filter for obtaining the low computation cycle and the energy efficient design of RIDFT algorithm.

### A. Input-Decimation Technique for RIDFT

The input tones are decimated in radix- $M$  first and hence the  $M$  decimated tones are combined with the corresponding twiddle factors (TWFs) with phase-difference  $\frac{\pi}{M}$  to form an aggregated tone that is fed into the recursive filter.

1) *Decimation-by- $M$* : The  $N$ -point IDFT with decimation-by- $M$  approach can be derived as

$$x_n = \sum_{k=0}^{N-1} X_k \cdot W_N^{-kn} = \sum_{k=0}^{\frac{N}{M}-1} F_{n,k} W_N^{-kn} \quad (1)$$

where  $n$  and  $k$  are the sample- and tone-index in the time- and frequency-domain, respectively.  $N$  and  $M$  are integers, where  $M$  is a factor of  $N$ . The  $F_{n,k}$  is defined as an aggregated tone and given as

$$F_{n,k} = X_k + X_{(k+\frac{N}{M})} \cdot W_N^{-n(\frac{N}{M})} + X_{(k+2\frac{N}{M})} \cdot W_N^{-k(2\frac{N}{M})} \\ + \dots + X_{[k+(M-1)\frac{N}{M}]} \cdot W_N^{-n[(M-1)\frac{N}{M}]} \quad (2)$$

Each aggregated tone is the summation of the  $M$  decimated tones with corresponding TWFs. The  $F_{n,k}$  is the kernel operation of the input-decimation technique.

2) *Derivation of Recursive Form*: It is clear that the original  $N$  input tones in (1) are shortened to  $N/M$  aggregated tones. In order to derive the recursive form, equation (1) can be reformulated as a convolution operation and given as

$$x_n = \sum_{k=\frac{N}{M}-1}^0 F_{n,(\frac{N}{M}-1-k)} \cdot W_N^{-(\frac{N}{M}-1-k)n} \\ = W_N^{-\frac{N}{M}n} \cdot \left[ \left( \sum_{k=0}^{\frac{N}{M}-1} F_{n,(\frac{N}{M}-1-k)} \cdot W_N^{kn} \right) \cdot W_N^n \right] \\ = W_M^{-n} \cdot \left[ \left( \dots \left( (F_{n,0} \cdot W_N^n + F_{n,1}) \cdot W_N^n + F_{n,2} \right) \cdot W_N^n \right. \right. \\ \left. \left. + \dots + F_{n,(\frac{N}{M}-2)} \cdot W_N^n + F_{n,(\frac{N}{M}-1)} \right) \cdot W_N^n \right] \quad (3)$$

where  $W_M^{-n}$  is defined as an output TWF. The difference equation inside the brackets of (3) is modelled as the recursive form and the intermediate output  $S_{n,k}$  is described as

$$S_{n,k} = (S_{n,k-1} + F_{n,k}) \cdot W_N^n, \quad k = 0, 1, \dots, \frac{N}{M} - 1 \quad (4)$$

where  $S_{n,-1} = 0$ . Hence, a single DFT output is acquired by sampling  $S_{n,k}$  at cycle time  $k = (\frac{N}{M} - 1)$  as well as multiplying with  $W_M^{-n}$ , and given as

$$x_n = W_M^{-n} \cdot S_{n,k} \Big|_{k=\frac{N}{M}-1} \quad (5)$$

Therefore, the computation cycle of a single DFT output is  $(\frac{N}{M} - 1)$  excluding that of input-decimation as derived in (2). In

view of the signal flow graph (SFG) formation, the  $z$ -domain transfer function (TF) of (5) can be derived as

$$H_n(z) = \frac{X_n(z) \cdot S_n(z)}{S_n(z) \cdot F_n(z)} = W_M^{-n} \cdot \left\{ \frac{W_N^n}{1 - W_N^{-n} z^{-1}} \right\} \\ = W_M^{-n} \cdot \left\{ \frac{W_N^n - z^{-1}}{1 - 2 \cos\left(\frac{2\pi n}{N}\right) z^{-1} + z^{-2}} \right\} \quad (6)$$

The equation inside the braces of (6) can be realized by a second order recursive filter. The SFG of (6) is shown in Fig. 2 (a) excluding the related paths of  $H_{N-n}(z)$ .

3) *Symmetric Property*: Based on the symmetric property on TWFs, the hardware complexity and the computation cycle of RIDFT can be simultaneously relaxed. For the **hardware-saving**, the symmetrical TF  $H_{N-n}$  can be derived as

$$H_{N-n}(z) = W_M^{-(N-n)} \cdot \left\{ \frac{W_N^{-n} - z^{-1}}{1 - 2 \cos\left(\frac{2\pi n}{N}\right) z^{-1} + z^{-2}} \right\} \quad (7)$$

The denominator inside the braces of (7) is the same as that of (6). Hence, the implementations of  $H_n(z)$  and  $H_{N-n}(z)$  can concurrently share the same feedback path. Then, both TFs can be merged into single SFG as shown in Fig. 2 (a).

For the **computation-cycle-shortening**, the aggregated tone of input-decimation RIDFT has the **symmetrical** feature describe as  $F_{N-n,k} = F_{n,k}$ , where it is true iff  $N$  is a power of 2 and  $n$  is even except for  $n = 0$  and  $N/2$ . Both  $x_n$  and  $x_{N-n}$  can be concurrently obtained at this situation. For  $N = 64$  and  $M = 4$ , the recursive filter can yield 23.4% reduction of the computation cycle.

4) *Discussion*: For the **computational complexity**, the decimation factor  $M$  can not be arbitrarily expanded to reduce the number of aggregated tones since the kernel operation of input-decimation will induce plenty of computational complexities. For  $N = 64$ , the computational complexity of the input-decimation as described in (2) for  $M = 2, 4, 8$  and 16 is illustrated in Fig. 2 (c). Apparently, there is no physical multiplication for  $M = 2$  and 4 because the TWFs are equal to  $\pm 1$  and  $\pm j$ . For the **computation cycle**, the recursive filter only requires  $N/M$  sequences to compute single IDFT output. Therefore, the reduction of computation cycle ( $\eta_{cycle}$ ) is  $(1 - \frac{1}{M}) \times 100\%$ , as shown in the bottom of Fig. 2 (c), without applying the symmetric property.

Although a large  $M$  is able to increase  $\eta_{cycle}$  of the recursive filter, requiring a large number of computations for the input-decimation kernel is a fatal issue. The selection of  $M$  is a trade-off between the computational complexity of the input-decimation kernel and the computation cycle of the recursive filter. Therefore,  $M = 4$  is appropriate to realize the input-decimation technique for RIDFT algorithm. Without loss of generality, the derivation of the input-decimation RIDFT can be completely applied to that of RDFT.

## III. HARDWARE DESIGN CONSIDERATION

### A. Architecture Design

The hardware architecture of the decimation-by-4 RIDFT is illustrated in Fig. 2 (b). For the beamforming application as

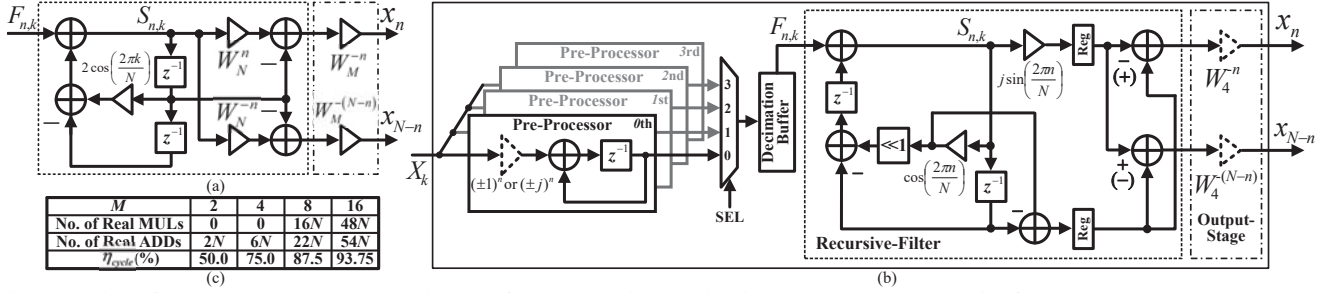


Fig. 2. (a) Signal flow graph and (b) hardware architecture of the proposed input-decimation RIDFT, where the recursive-filter and the output-stage are enclosed by dash- and dash-dot line, respectively. The dash-line triangles express the non-physical multiplications for  $\pm 1$  and  $\pm j$ . (c) Computational complexity of input-decimation kernel.

shown in Fig. 1, the input and the output sequences are the subcarrier decision error  $E_k$  in the frequency domain and the decision error sample  $e_n$  in the time domain, respectively. The related hardware considerations are described as below.

1) *Pre-Processor and Decimation-Buffer*: The pre-processor is employed to mainly perform decimation-by-4 operation and to generate the aggregated tone given as

$$F_{n,k} = X_k + X_{(k+\frac{N}{4})} \cdot W_N^{-\frac{N}{4}n} + X_{(k+\frac{N}{2})} \cdot W_N^{-\frac{N}{2}n} + X_{(k+\frac{3N}{4})} \cdot W_N^{-\frac{3N}{4}n} \quad (8)$$

where  $W_N^{-\frac{N}{4}n} = (j)^n$ ,  $W_N^{-\frac{N}{2}n} = (-1)^n$  and  $W_N^{-\frac{3N}{4}n} = (-j)^n$ . Due to the periodical property of TWFs, the aggregated tones also have the **periodical** feature described as  $F_{n,k} = F_{(n \bmod 4),k}$ . Therefore, for  $N = 64$ , it only requires 64 aggregated tones, such as  $F_{0,k}$ ,  $F_{1,k}$ ,  $F_{2,k}$  and  $F_{3,k} \forall k = 0, \dots, 15$ , to calculate all RIDFT outputs. There are four pre-processors to compute their own aggregated tones. All aggregated tones are stored in the decimation-buffer. The size of the decimation-buffer is  $N$  that is derived as  $(M \cdot \frac{N}{M})$ . All RIDFT outputs are acquired from  $M$ -group and each group has  $\frac{N}{M}$  aggregated tones.

2) *Recursive-Filter and Output-Stage*: The retiming registers denoted “Reg” as illustrated in the right side of the recursive filter are used to split the critical path. The critical period of the recursive filter is equal to  $T_m + 2T_a$ , where  $T_m$  and  $T_a$  express the computing times of the multiplier and the adder, respectively.

In order to reduce the hardware complexity, the module-folding and -sharing techniques are employed to decrease the number of multipliers in the recursive-filter. The TWFs ( $W_N^{\pm n}$ ) in the feedforward path of RIDFT can be decomposed as  $\cos(\cdot)$  and  $\sin(\cdot)$  terms, where the  $\cos(\cdot)$  term is identical to that of feedback path. Therefore, both  $\cos(\cdot)$  terms can be folded together. In view of an efficient hardware design, only one multiplier is realized to fulfill both  $\cos(\cdot)$ - and  $\sin(\cdot)$ -multiplication since the  $\sin(\cdot)$ -multiplication is only active at  $k = \frac{N}{4} - 1$ . For the output-stage, there is no physical multiplier because  $W_4^{-n}$  and  $W_4^{-(N-n)}$  are equal to  $\pm 1$  or  $\pm j \forall n$ .

### B. System Model and Physical Design

The system parameters for IEEE 802.11n high throughput mode with bandwidth of 20 MHz are described as follows:  $N$

(IDFT/DFT point) = 64;  $f_\Delta$  (subcarrier spacing) = 312.5 kHz;  $N_d/N_p$  (data/pilot subcarrier) = 52/4;  $T_u$  (IDFT/DFT duration) = 3.2  $\mu$ s;  $T_g$  (normal/short guard interval) = 0.8/0.4  $\mu$ s;  $T_s$  (OFDM symbol duration with normal/short guard interval) = 4.0/3.6  $\mu$ s; and  $T$  (sample duration in the time domain) = 0.05  $\mu$ s. Therefore, the execution time of RIDFT has to be less than 3.6  $\mu$ s no matter which guard interval (normal or short) is chosen. In order to meet the design target, the 8 partial outputs (POs) of RIDFT are used for updating the beamforming weight  $\hat{w}_m$  since its execution time is equivalent to 3.3  $\mu$ s.

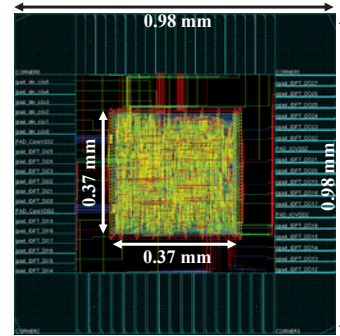


Fig. 3. Layout of input-decimation RIDFT.

The physical implementation for the input-decimation RIDFT with decimation-by-4 is also fulfilled with the standard cell-based design flow using Taiwan semiconductor manufacturing company (TSMC) 0.18  $\mu$ m CMOS general purpose process. The Verilog register transfer language (RTL) is synthesized by Design Compiler using the worst case with the slow-slow process corner and the supply voltage of 1.8 V to meet the timing constrained of 25 ns. The automatic placement & routing (APR) is realized by IC Compiler. The chip layout is shown in Fig. 3 and the physical design is summarized in Table I.

## IV. DISCUSSION AND COMPARISON

For the **computational complexity**, it is a realization indicator to demonstrate the computational feature for various RDFT/RIDFT algorithms as illustrated in Table II. It is well-known that the upper bound for computation cycle, real multiplication and addition for Goertzel [10] is  $N(N+1)$ ,  $2N(N+3)$  and  $4N(N+2)$ , respectively. For the input-decimation RIDFT with decimation-by-4, the quantities of

computation cycles, real multiplications and additions are reduced to the levels of  $3N^2/16$ ,  $3N^2/8$  and  $3N^2/4$ , respectively. For  $N = 64$ , the reduction percentages achieved by the input-decimation RIDFT algorithm can reach 78.4%, 80.6% and 77.8% for the computation cycle, real multiplication and addition, respectively, compared with [10].

TABLE I  
PHYSICAL DESIGN SUMMARY FOR INPUT-DECIMATION RIDFT.

<b>Process</b>	0.18 $\mu\text{m}$ CMOS GP Process
<b>Design</b>	Input-decimation 64-point RIDFT
<b>Application</b>	Beamforming tracking for OFDM receiver
<b>Wordlength (bit)</b>	13 (I/P), 14 (Internal-1 <sup>#</sup> ), 19 (Internal-2 <sup>#</sup> ), 14 (O/P)
<b>Execution time (<math>\mu\text{s}</math>)</b>	22.43 (FO <sup>Δ</sup> ), 3.3 (8-PO <sup>Δ</sup> )
<b>Area (<math>\text{mm}^2</math>)</b>	0.137
<b>Normalized gate count</b>	29.13 K
<b>Power consumption</b>	5.16 mW (FO) @40 MHz, 1.8 V

(#): The wordlength of pre-processor output, namely, the wordlength of decimation-buffer; ( $\Delta$ ): The internal wordlength of recursive filter; (A): The full output (FO) samples, namely, 64 output samples; (P): The partial output (PO) samples, e.g., 8-PO = 8 output samples;

TABLE II  
COMPUTATIONAL COMPLEXITY

Design	Computation Cycle	Real MUL	Real ADD
Goertzel [10]	$N(N+1)$	$2N(N+3)$	$4N(N+2)$
RDFT [3]	$N^2/2$	$2N(N+3)$	$4N(N+2)$
RDFT [5]	$(N/2-1)(N+1)$	$(N+1)(N-2)$	$N(4N+14)-4$
Muticycle Dual-RDFT [6]	$(c+1)N+m(m+1)$	$2N(m+c+2)$	$4N(m+c+2)$
Multi-RDFT [7]	PF <sup>2</sup> $(Nc+N+2c)/2 + m(m+3)/2-2$	$N(c+1)+m(N+c)$	$2N(c+2)+2m(N+c)$
	CF <sup>3</sup> $(Nc+N+2c)/2 + m(m+3)/2-5m+1$	$N(c+m+12)-4$	$2N(c+m+5.5)-2$
Folded-RDFT [8]	PF <sup>4</sup> $\Delta$	$2m(c+1)(c/2-1) + c(m+1)(m-1)$	$4N(c/2-1)+4c + 4N(m+1)/2+4m$
This work ( $M=4$ )	$(3N/4+1)(N/4+1)+N$	$(3N/4+1)(N/2+2)$	$(3N/4+1)(N+4) + (N/2-2)+6N$
<sup>#</sup> Reduction efficiency $\eta_1/\eta_2$ ( $N=64$ )	56.2%/55.5%	80.6%/58.7%	77.8%/78.3%

( $\delta$ ): The  $\eta_1/\eta_2$  is the reduction efficiency of the proposed algorithm compared with [3]/[5]; ( $\Delta$ ): It is NOT reported in [8];

TABLE III  
HARDWARE RESOURCE

Design	MULs	ADDs	Storage			Critical Period
			I/P	Temporal/Buffer	Coef. ROM	
RDFT [3]	10	17	<sup>1</sup> 318 $\times$ 24b	0	<sup>1</sup> 318-word	$T_m+2T_a$
RDFT [5]	2	13	<sup>E</sup> 288 $\times$ 32b	0	<sup>2</sup> 1301-word	$T_m+2T_a$
Muticycle Dual-RDFT [6]	4	8	<sup>E</sup> 288 $\times$ 32b	<sup>B</sup> 4 $\times$ 11-word	<sup>3</sup> 694-word	$T_m+2T_a$
Multi-RDFT [7]	6	14	<sup>E</sup> 480 $\times$ 32b	<sup>B</sup> 8 $\times$ 20-word	<sup>H</sup> 0	$T_m+2T_a$
Folded-RDFT [8]	2	4	0	<sup>T</sup> 2 $\times$ 480 $\times$ 32b	<sup>H</sup> 0	$T_m+2T_a$
Hybrid Arch. [9]	4	10	0	<sup>T</sup> 480 $\times$ 32b <sup>B</sup> 30 $\times$ 32b	<sup>H</sup> 0	$T_m+3T_a$
This work	<sup>\delta</sup> 1	18	0	<sup>H</sup> 64 $\times$ 28b	<sup>H</sup> $\Delta$ 0	$T_m+2T_a$

(E): Exclude this item in design; (I): Include this item in design; (T): Use for input and temporal storage; (U): Only use for temporal storage; (B): Data buffer; (H): Hard-wired coefficient; ( $\delta$ ): The constant multiplier including SD coefficients; ( $\Delta$ ): The SD coefficients are merged into the constant multiplier; (1): Coefficients for 212- and 106-point; (2): Coefficients for 288-, 256-, 176-, 112-, 212-, 165- and 106-point; (3): Coefficients for 288-, 256-, 176- and 112-point;

In view of an impartial comparison in the reduction efficiency, only single RDFT module designs [3][5] are considered as the objects of comparison. For  $N = 64$ , the input-decimation RIDFT algorithm can reach 56.2%/55.5%, 80.6%/58.7% and 77.8%/78.3% of reduction efficiency ( $\eta_1/\eta_2$ ) in the computation cycle, real multiplication and addition, respectively, compared with [3]/[5].

For the **hardware resource**, it is also a realization indicator to exhibit the implementation cost for various RDFT/RIDFT architectures as shown in Table III. The input-decimation

RIDFT only requires one multiplier, which is the least number of multipliers compared with the others. The multiplier is realized by a constant multiplier [12] combined with TWFs in place of a physical multiplier and a coefficient ROM. The pre-processing and the recursive filter require 8- and 10-adder, respectively.

The input storage is also a crucial component that stores the input sequences to iteratively calculate the required outputs of recursive filter. It is meaningful to include the input storage in their designs [3][5][6][7]. The input-decimation RIDFT does not require input storage, but it need a temporal storage to keep the aggregated tones for computing the required outputs of recursive filter. Also, it is significant to have a coefficient ROM [3][5][6] for storing TWFs including  $\cos(\cdot)$  and  $\sin(\cdot)$ . The hard-wired realizations [7][8][9] for TWFs are an alternative approach in place of the coefficient ROM. The critical period for various RDFT/RIDFT architectures is shown in the right-side of Table III. The critical period for the hybrid architecture [9] is equivalent to  $T_m + 3T_a$ . Besides, the critical period for the other designs is equal to  $T_m + 2T_a$ .

For the **performance comparison** of physical design for various RDFT/RIDFT implementations as shown in Table IV, the normalized area, the DFTs/Energy and the computational efficiency per unit area (CEUA) [9] are used to examine these physical designs, and described as

$$\text{Normalized Area} = \frac{\text{Area}}{(\text{Tech.}/0.18 \mu\text{m})^2} \quad (9)$$

$$\frac{\text{DFTs}}{\text{Energy}} = \frac{(\text{Tech.}/0.18 \mu\text{m}) \times (\text{Volt.}/1.8 \text{ V})^2}{\text{Power} \times \text{Exec. Time}/(\text{DFT point})^2 \times 10^3} \quad (10)$$

$$\text{CEUA} = \frac{\text{DFTs}/\text{Energy}}{\text{Normalized Area}} \quad (11)$$

In order to do a fair comparison, both technology and supply voltage are normalized by 0.18  $\mu\text{m}$  and 1.8 V, respectively, for different technologies and supply voltages. For unifying the calculation of DFTs/Energy, the execution time is also normalized by the square of DFT-point regardless of the dual-RDFT [6] (2-RDFT kernels working in parallel processing), the multi-RDFT [7] (3-RDFT kernels operating in two-stage processing and working in parallel processing at the second stage), the folded-RDFT [8] (an iterative operation of single RDFT kernel) and the hybrid architecture [9] (a configurable hardware module used for radix-2<sup>2</sup> FFT and RDFT).

Obviously, the input storage is not involved in these realizations [5][6][7] so that the practical chip area and power consumption for these designs are more than those as shown in Table IV, which means that both DFTs/Energy and CEUA will be smaller. From DFTs/Energy point of view, for  $N = 256$ , the multi-RDFT [7] and the hybrid architecture [9] are around 1.7 and 2.6 times better than that of the proposed input-decimation RIDFT. The main reason for [7] is that it has 3-RDFT kernels inside, where the first one is in the first-stage and the other two RDFTs work in parallel in the second-stage. The primary reason for [9] is that it is not a pure RDFT design. For  $N = 256$ , the hybrid architecture is only configured as radix-2<sup>2</sup> single-path delay feedback (SDF) FFT to perform entire operation

TABLE IV  
PERFORMANCE COMPARISON OF PHYSICAL DESIGN FOR VARIOUS RDFT/RIDFT ARCHITECTURES

Design	[3]	[5]	[6]	[7]	[8]	[9]	This work
Algorithm	RDFT	RDFT	Muticycle Dual-RDFT	Multi-RDFT	Folded-RDFT	Hybrid Architecture	Input-Decimation RIDFT
Application	DTMF	DRM/DTMF	DRM	DRM	DRM/DRM <sup>+</sup>	DRM/DRM <sup>+</sup>	Beamforming for OFDM Rx.
Technology	0.13 $\mu\text{m}$	0.18 $\mu\text{m}$	0.18 $\mu\text{m}$	0.18 $\mu\text{m}$	0.18 $\mu\text{m}$	0.18 $\mu\text{m}$	0.18 $\mu\text{m}$
DFT ( $N$ -pt)	212, 106	288, 256, 176, 112, 212, 165, 106	288, 256, 176, 112	288, 256, 176, 112, 480, 60	288, 256, 176, 112, 480, 60	288, 256, 176, 112, 480, 60, 27	64
Operating Clock (MHz)	20	25	25	25	25	25	40
Execution Time	1.12 ms (212-pt) 280.9 $\mu\text{s}$ (106-pt)	1.65 ms (288-pt) 1.31 ms (256-pt)	384 $\mu\text{s}$ (288-pt) PF(c=32, m=9) 1.31 ms (256-pt)	193.68 $\mu\text{s}$ (288-pt) PF(c=32, m=9) 113.08 $\mu\text{s}$ (288-pt) CF(c=18, m=16) 90.6 $\mu\text{s}$ (256-pt) CF(c=16, m=16)	267.72 $\mu\text{s}$ (288-pt) PF(c=32, m=9)	180.52 $\mu\text{s}$ (288-pt) CF(c=16, m=18) 80.52 $\mu\text{s}$ (256-pt) CF(c=256, m=0)	22.43 $\mu\text{s}$ (A)FO 3.3 $\mu\text{s}$ (8-PO)
Power-Consumption/Supply-Voltage	$\Delta\beta$ 1.25 mW/1.2 V	$\#\beta$ 5.98 mW/1.98 V	$\#\beta\Omega$ 8.44 mW/1.98 V	$\#\Omega$ 14.6 mW/1.98 V	$\Omega$ 9.62 mW/1.7 V	$\Omega$ 8.8 mW/1.8 V	$\Phi\Omega$ 5.16 mW/1.8 V
Storage	Input: $\Omega$ 318x24b Temporal/Buffer: 0	$\Omega$ 288x24b	$\Omega$ 288x32b 4x11-word	$\Omega$ 480x32b 8x20-word	0 $\Gamma$ 2x480x32b	0 $\Gamma$ 480x32b + $\beta$ 30x32b	0 $\Upsilon$ 64x28b
Core Area (mm <sup>2</sup> )	$\Delta\beta$ 0.182 $\Delta\beta$ 0.348	$\#\beta$ 0.154	$\#\beta\Omega$ 0.265	$\#\Omega$ 0.705	$\Omega$ 0.714	$\Omega$ 0.436	$\Phi\Omega$ 0.137
Normalized Area (mm <sup>2</sup> )	$\Delta\beta$ 0.348	$\#\beta$ 0.154	$\#\beta\Omega$ 0.265	$\#\Omega$ 0.705	$\Omega$ 0.714	$\Omega$ 0.436	$\Omega$ 0.137
DFTs/Energy	$\Delta\beta$ 10.3 M (212-pt) $\Delta\beta$ 10.3 M (106-pt)	$\#\beta$ 10.2 M (288-pt) $\#\beta$ 10.1 M (256-pt)	$\#\beta\Omega$ 31.0 M (288-pt) $\#\beta\Omega$ 7.2 M (256-pt)	$\#\Omega$ 35.5 M (288-pt/PF) $\#\Omega$ 60.8 M (288-pt/CF) $\#\Omega$ 60.0 M (256-pt/CF)	$\Omega$ 28.7 M (288-pt)	$\Omega$ 52.2 M (288-pt) $\Omega$ 92.5 M (256-pt)	$\Phi\Omega$ 35.4 M (FO)
CEUA	$\Delta\beta$ 30 M (212-pt) $\Delta\beta$ 30 M (106-pt)	$\#\beta$ 66 M (288-pt) $\#\beta$ 66 M (256-pt)	$\#\beta\Omega$ 117 M (288-pt) $\#\beta\Omega$ 27 M (256-pt)	$\#\Omega$ 50 M (288-pt/PF) $\#\Omega$ 86 M (288-pt/CF) $\#\Omega$ 85 M (256-pt/CF)	$\Omega$ 40 M (288-pt)	$\Omega$ 120 M (288-pt) $\Omega$ 212 M (256-pt)	$\Phi\Omega$ 258 M (FO)

(A): The full output (FO) samples, namely, 64 output samples; (P): The partial output (PO) samples, e.g., 8-PO = 8-output samples; (D): The prime-factor (PF); (E): The common-factor (CF); (F): Neither PF- nor CF-decomposition; (Q): Only use for input storage; (T): Use for input and temporal storage; (U): Only use for temporal storage; (B): Data buffer; (#): Exclude the input storage; ( $\Delta$ ): Include input storage; ( $\Omega$ ): Include temporal/buffer storage; ( $\beta$ ): Include coefficient storage; ( $\Phi$ ): The power-consumption estimated for 64-output samples;

and the execution time is proportional to  $\frac{N}{4} \log_4 N$ , not  $N^2$ . Finally, in view of the hardware computational efficiency of unit area cost, the proposed input-decimation RIDFT is able to reach 258 million of CEUA and really outperform to the previous designs [3][5]–[9].

## V. CONCLUSION

In this paper, the input-decimation RIDFT with decimation-by-4 is primarily presented for the weight updating of beamforming tracking for OFDM receiver. The input-decimation approach not only shortens the number of computation cycles but also reduces the number of computational complexities for RIDFT algorithm. The computation cycle of the proposed RIDFT algorithm is really decreased so that it can yield at least 55.5% reduction of total computation cycles compared with the considered algorithms. Besides, taking advantages of the input-decimation technique, the computational complexities of real multiplication and addition are reduced to 41.3% and 22.2%, respectively. Finally, the execution time of the proposed design can reach several  $\mu\text{s}$ -level, such as 3.3  $\mu\text{s}$  with 8 partial outputs. The physical implementation results show that the core area is 0.137 mm<sup>2</sup> with 0.18  $\mu\text{m}$  CMOS process. The power consumption is 5.16 mW with the supply voltage of 1.8 V and the operating clock of 40 MHz. The proposed design achieves 258 million of CEUA and is indeed superior to the previous works.

## REFERENCES

- [1] ITU Blue Book, Recommendation Q. 23: Multi-frequency Push-Bottom Signal Reception, Geneva, Switzerland, 1989.
- [2] L. D. Van and C. C. Yang, "High-speed area-efficient recursive DFT/IDFT architectures," *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, vol. 3, pp. 357–360, May 2004.
- [3] L. D. Van, C. T. Lin and Y. C. Yu, "VLSI Architecture for the Low-Computation Cycle and Power-Efficient Recursive DFT/IDFT Design," *IEICE Trans. on Fundam. Electron., Commun. Comput. Sci.*, vol. E90-A, no. 8, pp. 1644–1652, Aug. 2007.
- [4] Digital Radio Mondiale: System Specification, ETSI, ES 201 980 V2.1.1, Nov. 2003.
- [5] S. C. Lai, S. F. Lei, C. L. Chang, C. C. Lin and C. H. Luo, "Low Computational Complexity, Low Power, and Low Area Design for the Implementation of Recursive DFT and IDFT Algorithms," *IEEE Trans. on Circuits and Systems – II: Express Briefs*, vol. 56, no. 12, pp. 921–925, Dec. 2009.
- [6] S. C. Lai, W. H. Juang, C. L. Chang, C. C. Lin, C. H. Luo and S. F. Lei, "Low-Computation-Cycle, Power-Efficient, and Reconfigurable Design of Recursive DFT for Portable Digital Radio Mondiale Receiver," *IEEE Trans. on Circuits and Systems – II: Express Briefs*, vol. 57, no. 8, pp. 647–651, Aug. 2010.
- [7] S. C. Lai, W. H. Juang, Y. S. Lee and S. F. Lei, "High-Performance RDFT Design for Applications of Digital Radio Mondiale," *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pp. 2601–2603, May. 2013.
- [8] S. C. Lai, Y. S. Lee and S. F. Lei, "Low-Power and Optimized VLSI Implementation of Compact Recursive Discrete Fourier Transform (RDFT) Preprocessor for the Computations of DFT and Inverse Modified Cosine Transform (IMDCT) in Digital Radio Mondiale (DRM) and DRM<sup>+</sup> Receiver," *J. of Low Power Electron. and Appl.*, vol. 3, no. 2, pp. 99–113, May. 2013.
- [9] S. C. Lai, W. H. Juang, Y. S. Lee, S. H. Chen, K. H. Chen C. C. Tsai and C. H. Lee, "Hybrid Architecture Design for Calculating Variable-Length Fourier Transform," *IEEE Trans. on Circuits and Systems – II: Express Briefs*, vol. 63, no. 3, pp. 279–283, Mar. 2016.
- [10] G. Goertzel, "An Algorithm for the Evaluation of Finite Trigonometric Series," *Amer. Math Mon.*, vol. 65, pp. 34–35, Jan. 1958.
- [11] C. F. Wu, C. H. Chen, and M. T. Shiu, "Decision-Directed Beamforming and Channel Equalization Algorithm for IEEE 802.11n OFDM Systems," *IEEE Int. Symp. on Computer, Consumer and Control (IS3C)*, pp. 220–223, Jul. 2016.
- [12] T. Y. Chen, Y. H. Lin, C. F. Wu and C. K. Wang, "Cost-Efficient Design and Fixed-Point Analysis of IFFT/FFT Processor Chip for OFDM Systems," *Proceeding of 2011 Intern. Symp. on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–4, Apr. 2011.